**Evoza - Web Browser**

Biplov Gautam, Kalina Shrestha, Prochorus Rai, Rohit Shah, Sabin Raj Pokharel

BSc. (Hons) Computer Science with AI

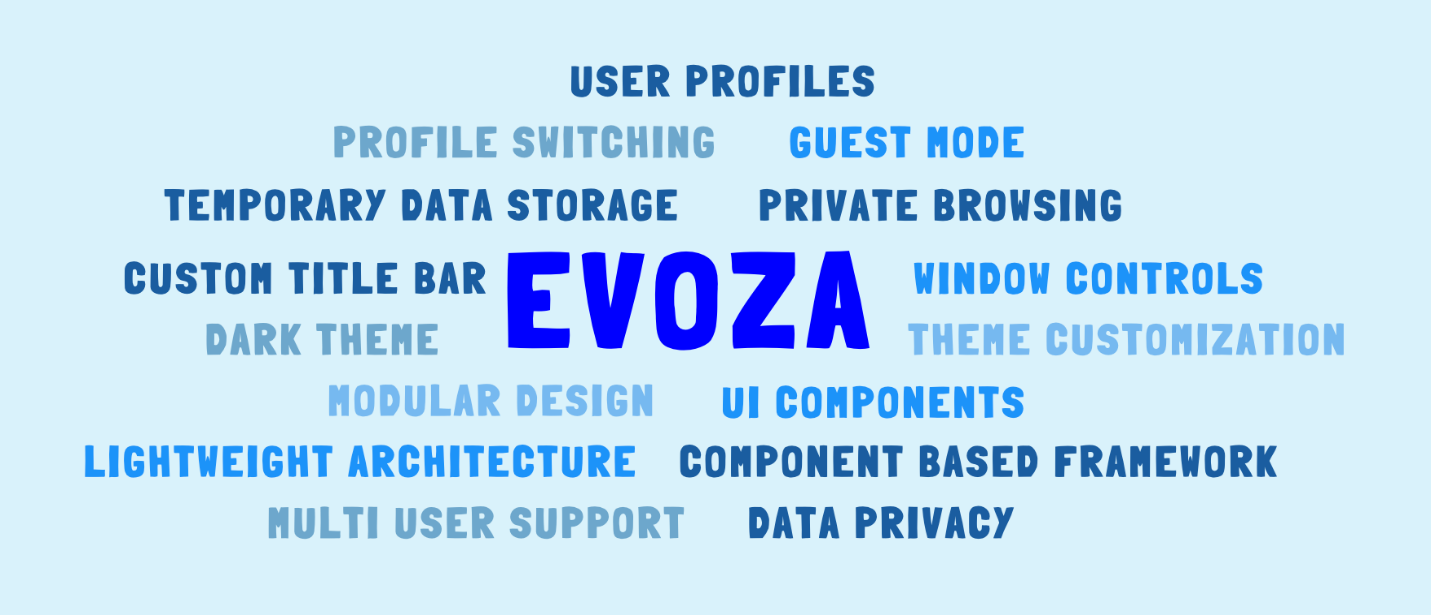
STW4009CEM Computing Activity Led Learning Project 2

Albert Maharjan

December 8, 2024

**Keywords**

* User Profiles
* Profile Switching
* Guest Mode
* Temporary Data Storage
* Private Browsing
* Custom Title Bar
* Password Reset
* Dark Theme
* Verification Email
* Modular Design
* UI Components
* Lightweight Architecture
* Component-Based Framework
* Multi-User Support
* Data Privacy



**Table of Contents**

**[Introduction](#_Toc184644786)** [6](#_Toc184644786)

**[Aim](#_Toc184644787)** [7](#_Toc184644787)

**[Objectives](#_Toc184644788)** [7](#_Toc184644788)

**[Problem Statement](#_Toc184644789)** [8](#_Toc184644789)

**[Data and Research](#_Toc184644790)** [10](#_Toc184644790)

**[Features](#_Toc184644791)** [11](#_Toc184644791)

**[Functionalities](#_Toc184644792)** [12](#_Toc184644792)

**[Functional Requirements](#_Toc184644793)** [12](#_Toc184644793)

**[Non-Functional Requirements](#_Toc184644794)** [13](#_Toc184644794)

**[Methodology](#_Toc184644795)** [14](#_Toc184644795)

**[Tools and Technology](#_Toc184644796)** [21](#_Toc184644796)

**[Conceptual diagram](#_Toc184644797)** [23](#_Toc184644797)

**[Prototype](#_Toc184644798)** [24](#_Toc184644798)

**[Scope](#_Toc184644799)** [25](#_Toc184644799)

**[SWOT Analysis](#_Toc184644800)** [25](#_Toc184644800)

**[Conclusion](#_Toc184644801)** [27](#_Toc184644801)

**[References](#_Toc184644802)** [28](#_Toc184644802)

**Table of Figures**

[Figure 1: 9](#_Toc184644717)

[Figure 2: 11](#_Toc184644718)

[Figure 3: 14](#_Toc184644719)

[Figure 4: 16](#_Toc184644720)

[Figure 5: 16](#_Toc184644721)

[Figure 6: 17](#_Toc184644722)

[Figure 7: 18](#_Toc184644723)

[Figure 8: 19](#_Toc184644724)

[Figure 9: 22](#_Toc184644725)

[Figure 10: 23](#_Toc184644726)

[Figure 11: 26](#_Toc184644727)

# **Introduction**

With the advancement in tools and technologies in this era, the internet browsing experience is also continually evolving to meet the diverse needs of users and make one’s life comfortable along with saving time. Today, to fulfill the varying requirements, users opt for fast, secure and highly personalized browsing tools [Kumar (2024)](#Bookmark1). Evoza , a cutting-edge web browser, is designed to address these demands and provide an efficient, feature-rich solution for seamless and secure browsing.

Keeping user-friendly interface with features like multiple profile support with customizable avatars, our browser is designed to suit the preference of each web user. Saving bookmarks and histories is made easier without having to be logged-in to the browser since the Guest Mode enables users for temporary browsing and Private/Incognito Mode guarantees complete privacy by preventing the storage of browsing data.

Evoza offers a highly adaptable window interface, featuring a custom title bar that integrates minimize, maximize, and close buttons for streamlined window management. It also boasts a dynamic dark theme, ensuring an aesthetically pleasing and user-friendly experience. Guaranteeing flexibility and adaptability, Evoza's modular design is crafted with separate UI components that seamlessly integrate core features. This approach ensures a delightful, comprehensive, and personalized web browsing experience, allowing users to customize their interface while maintaining a cohesive and efficient design. Each component is designed to be intuitive, ensuring smooth functionality and adaptability to various user preferences and workflows.

# **Aim**

The “Evoza” web browser is developed for providing a modern, secure and adaptable surfing experience that enhances efficiency, privacy, personalization and flexibility to meet diverse user requirements.

# **Objectives**

* **Personalization**: Features seamless switching between profiles and customized avatar, ensuring a sleek browsing experience.
* **Privacy and Security**: Modes like Guest Mode, Incognito/ Private Mode for user privacy allowing browsing the internet without leaving a trace.
* **Customizability**: Versatile interface featuring dark theme, custom title bar and modular design suitable as per individual preferences.
* **Efficiency**: Smooth browsing experience with easy access to bookmarks and histories to enhance user workflow.
* **Adaptability**: User friendly modular browser evolving with user needs and technological advancements.

# **Problem Statement**

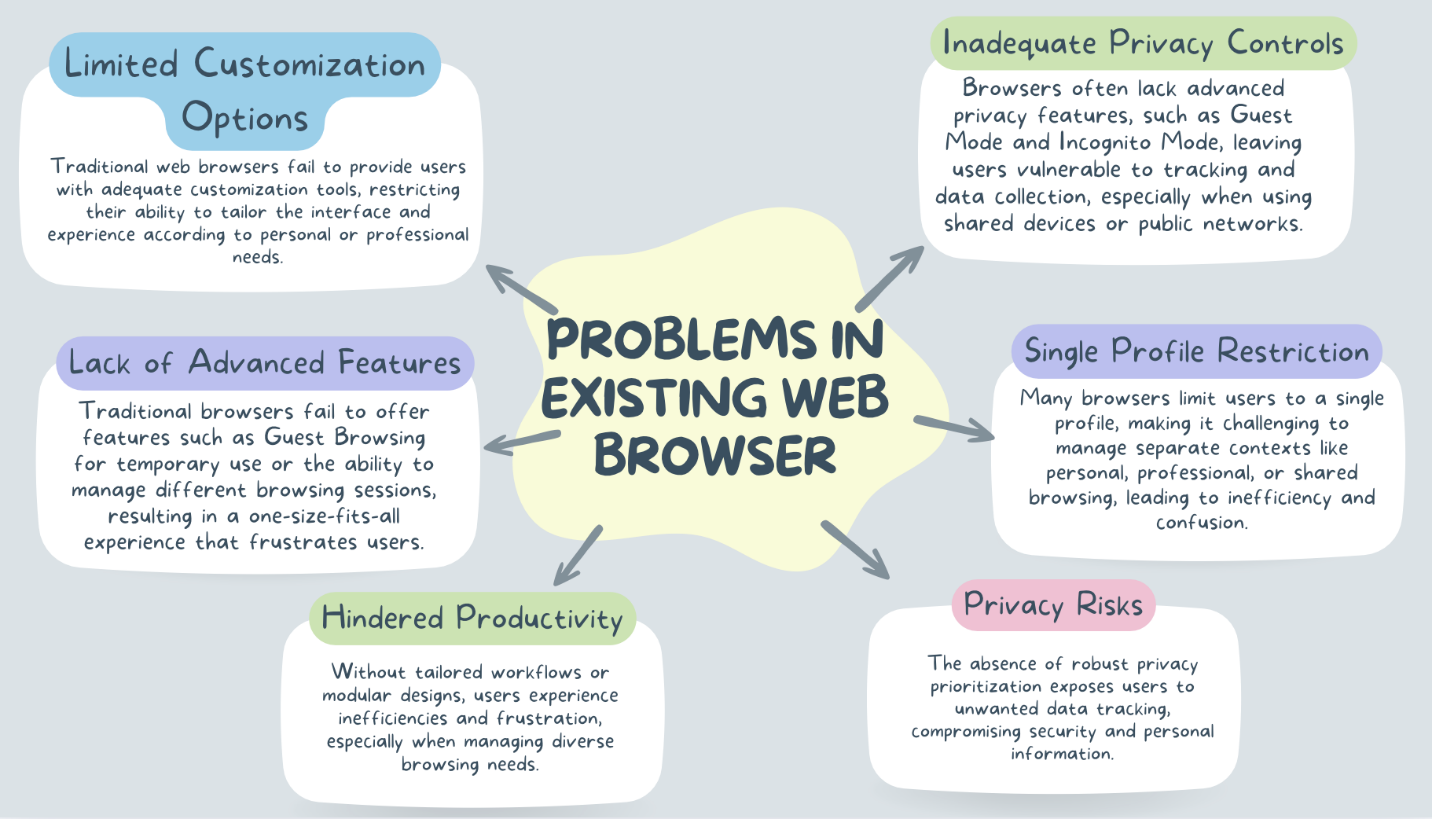
As the internet becomes an integral part of daily life, managing browsing experience has become an increasing challenge that users face today. Due to limited customization options, insufficient privacy settings, and poor support for handling various surfing contexts, traditional web browsers frequently fail to evolve. Many browsers even limit users to a single profile making it challenging to balance personal and professional sessions leading to confusion and disorganization.

Furthermore, users are subjected to unearned tracking and data collection as the traditional browsers often fall short to prioritize privacy. These browsers typically fail to offer robust privacy features, leaving personal data vulnerable to third-party tracking. Additionally, they lack the customization that surfers require in order to adjust their browsing experience based on requirements, whether it be for educational, professional or recreational purposes. As a result, users struggle to keep their browsing experience secure and compromise their privacy and overall experience.

To address these challenges, Evoza aims to provide a complete solution that integrates privacy, performance, and personalization. The browser allows users to customize their environment to meet their specific needs with features like multiple user profiles, Guest Mode for temporary surfing, Private/Incognito Mode for secure browsing, and a fully customizable dark theme. Its modular design offers improved privacy and protection along with flexibility, scalability, and a future-proof experience. Evoza gives consumers the ability to effectively control their surfing requirements while guaranteeing security, privacy, and easy personalization for a better online experience [Olsen (2022)](#Bookmark2).

Figure 1:

Problems in existing web browser



# **Data and Research**

Our research identified two relevant case studies: Google Chrome and Vivaldi Browser. Both browsers offer advanced features and are widely recognized for their unique approaches to web browsing.

Google Chrome is one of the most popular browsers globally, known for its speed, reliability, and extensive ecosystem of extensions and integrations. Its user-friendly design and seamless synchronization across devices set a high standard for modern browsing. However, Chrome’s limitations in personalization, lack of advanced privacy controls, and a single-profile system often leave users with an uncustomizable and less secure experience, especially for those requiring separate browsing contexts or enhanced privacy [GeeksforGeeks (2024)](#Bookmark3).

Vivaldi Browser, on the other hand, caters to users seeking advanced customization and productivity features. It offers options like adjustable tab management, a customizable interface, and tools such as built-in notetaking and web panels. However, it falls short in offering advanced privacy features like Guest Mode for shared browsing and multi-profile systems for managing personal and professional contexts seamlessly [Sean (2024)](#Bookmark4).

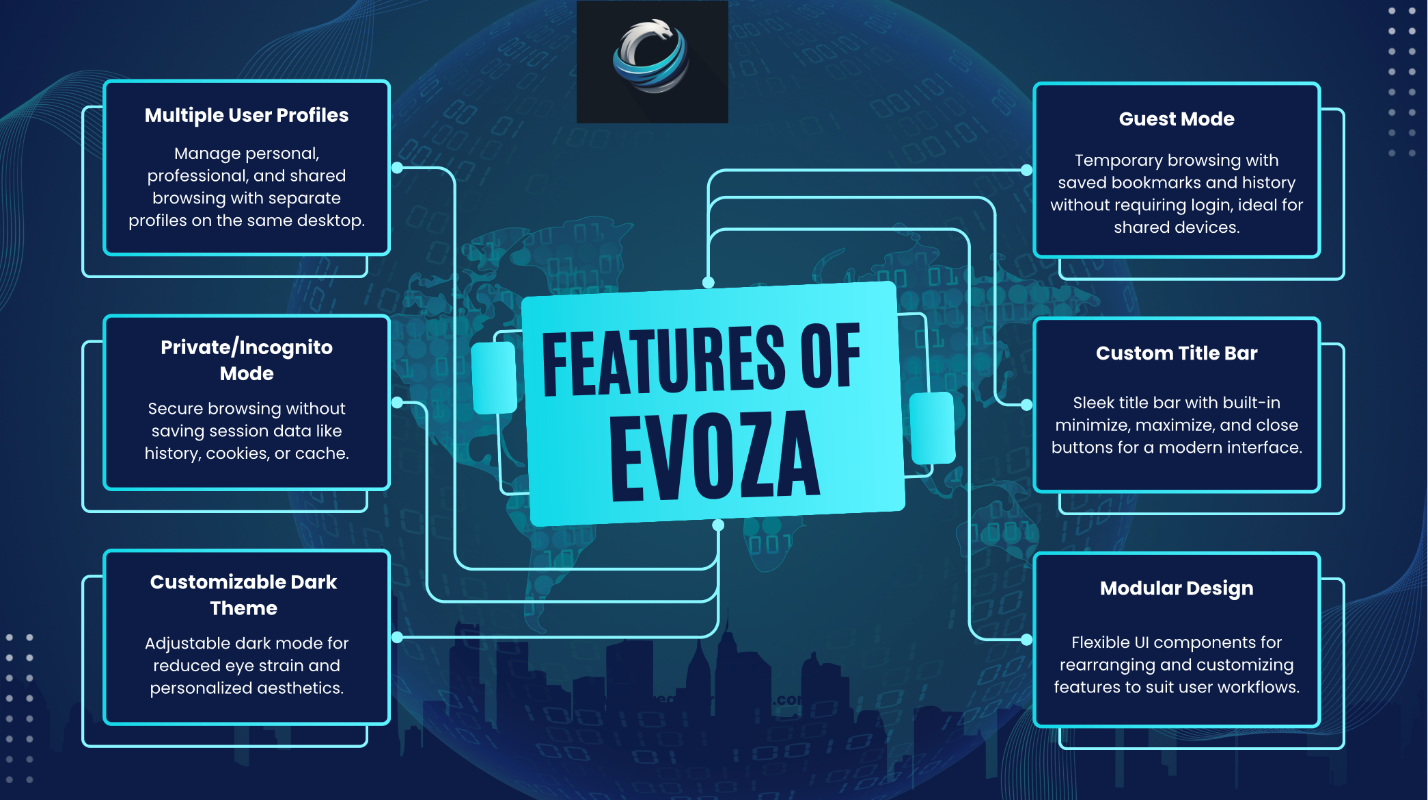
By analyzing these browsers, we identified the market gap: a need for a browser that combines Chrome’s simplicity and speed with Vivaldi’s customizability, while addressing critical privacy and usability gaps. Our browser, Evoza, incorporates learnings from these case studies, offering features like multi-profile support, Guest Mode, Private/Incognito Mode, and a modular design for complete user control. With these innovations, Evoza aims to redefine the web browsing experience by balancing privacy, performance, and personalization in a user-friendly and future-proof platform.

# **Features**

* Different profiles for different users in same desktop
* Signup for adding profiles & logging in using exiting profiles
* Guest / Private mode to browse without saving data
* Custom title bar, custom popup alert messages and mnay more
* Customizable dark theme
* Modular design with separate UI components

Figure 2:

Features of Evoza



# **Functionalities**

## **Functional Requirements**

Any requirements that specify what the system must be able to do are known as functional requirements. It consists of things like authorization levels, external interfaces, administrative functions, authentication, and business rules [(Gate Smashers, 2022)](#Bookmark5).

|  |  |  |
| --- | --- | --- |
| Requirement ID | Functional Requirement | Description |
| FR-01 | Multiple User Profiles | Enables smooth transitions between profiles with separate bookmarks , histories, settings and preferences. |
| FR-02 | Avatar Customization for Profiles | Avatar customization makes user stand distinct from others and giving the browsing experience a unique feel. |
| FR-03 | Bookmark Management | Users may save bookmarks for effective management. |
| FR-04 | Browsing History Management | Users can access the history and selectively manage it, maintaining privacy and convenience. |
| FR-05 | Guest Mode | Guest mode enables browsing without logging in |
| FR-06 | Private/Incognito Mode | No personal data is stored after the session ends. |
| FR-07 | Customizable Dark Theme | Dark theme option for comfortable browsing experience. |
| FR-08 | Custom Title Bar | The title bar is customizable with functionality of minimize, maximize, and close buttons. |

## **Non-Functional Requirements**

Any requirements that specify how a system must carry out specific tasks are known

as non-functional requirements. It includes all other requirements such as performance,

throughput, response time, capacity, scalability, security, maintainability, etc. that are not addressed by the functional requirements [(Gate Smashers, 2022)](#Bookmark5).

|  |  |  |
| --- | --- | --- |
| Requirement ID | Non-Functional Requirement | Description |
| NFR-01 | Performance | The browser should load websites quickly ensuring smooth performance. |
| NFR-02 | Security | It must provide encryption, protect against phishing, and secure browsing data. |
| NFR-03 | Usability | The interface should be simple and intuitive for users’ comfort. |
| NFR-04 | Compatibility | The browser should work seamlessly across multiple operating systems (Windows, macOS, Linux). |
| NFR-05 | Scalability | It should efficiently handle increasing users, profiles, and sessions without sacrificing speed or performance. |
| NFR-06 | Availability | The browser should have minimal downtime, with automatic updates to ensure stability, security, and a consistent user experience. |

# **Methodology**

The project will employ the agile development methodology since it allows for an

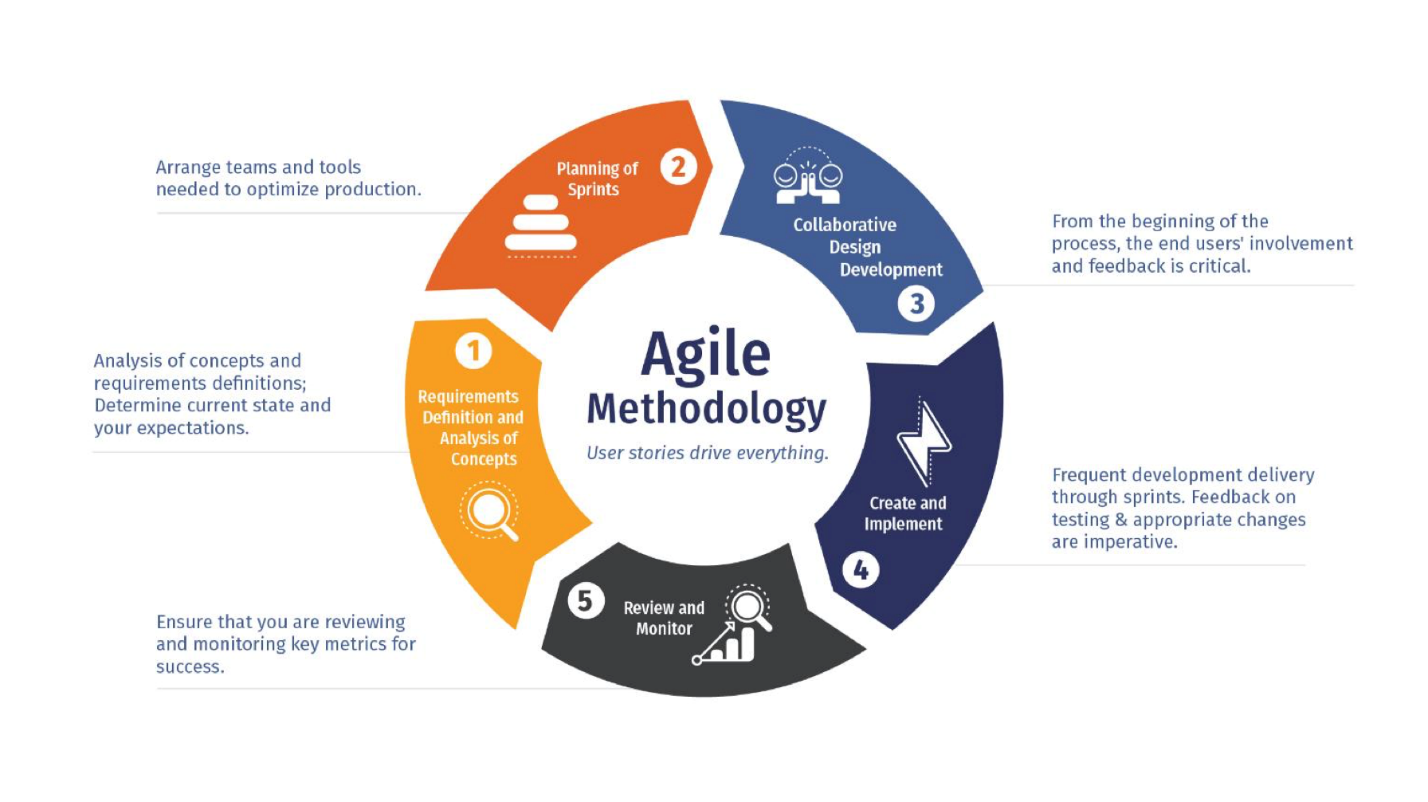
iterative approach with continuous delivery and integration of evolving requirements over the

course of development [Laoyan (2024)](#Bookmark6). Because the objectives of our project align with the

core principles of agile, this methodology was selected.

Figure 3:

Agile Methodology

 The Agile methodology is an approach for project management that prioritizes

continuous improvement and cross-functional cooperation. It divides tasks into dynamic,

iterative stages called sprints, during which teams plan, carry out, and assess their work in

turn. Teams evaluate their performance and approach following each sprint, adjusting as

needed for the following one to guarantee continuous development and adaptation.

We will be following an agile development methodology using a mix of Scrum and

Kanban frameworks i.e. Scrumban which is optimal for a flexible evolving system requiring

continuous delivery.

Figure 4:

Scrum Framework at a glance

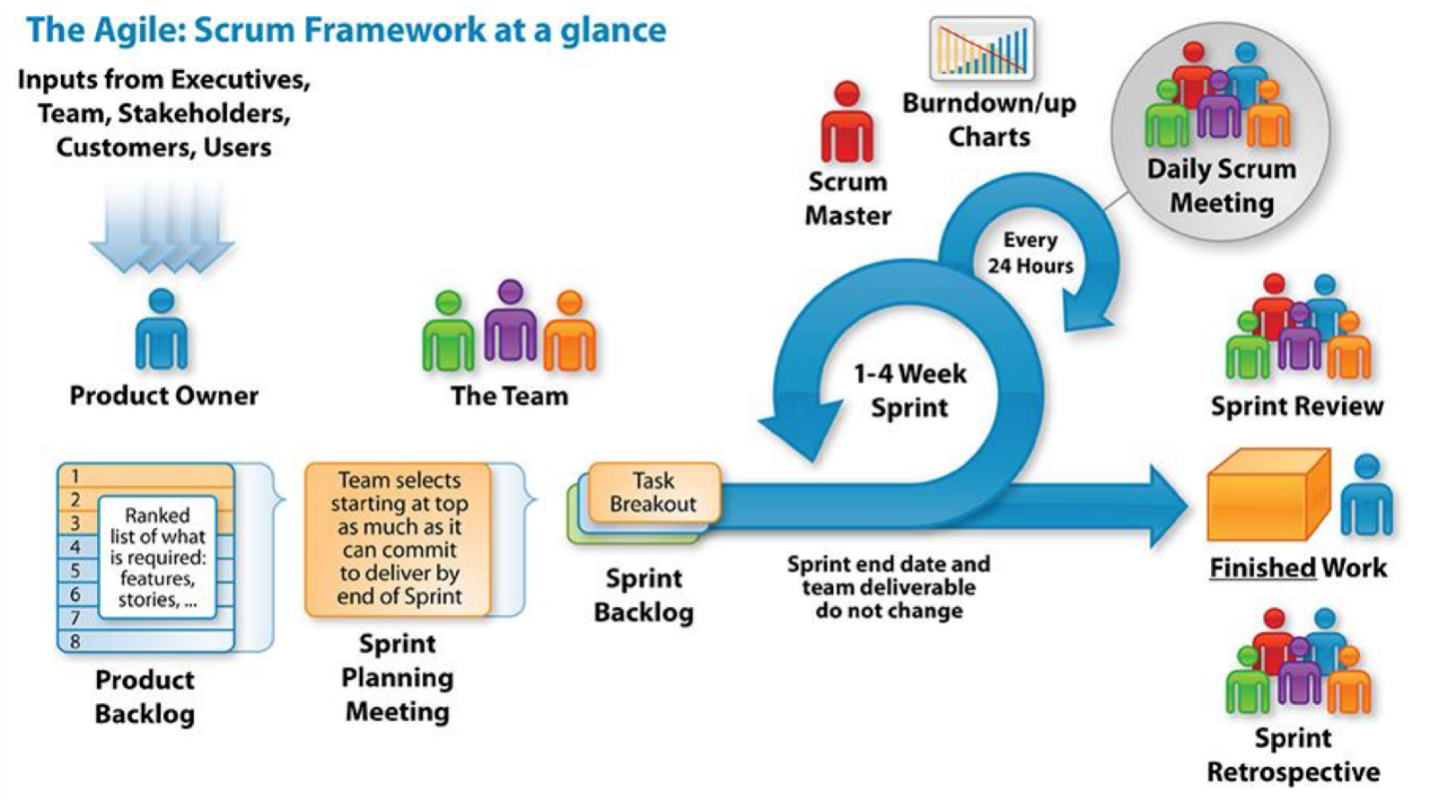


Figure 5:

Scrum Process

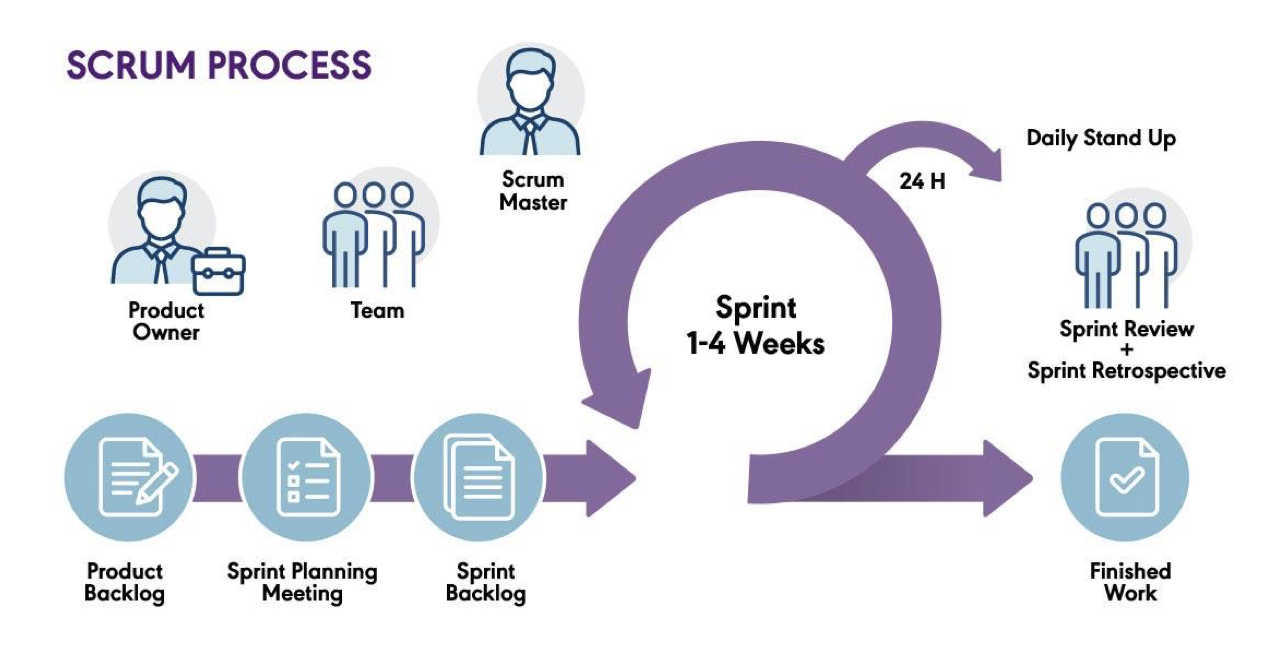


Figure 6:

Kanban

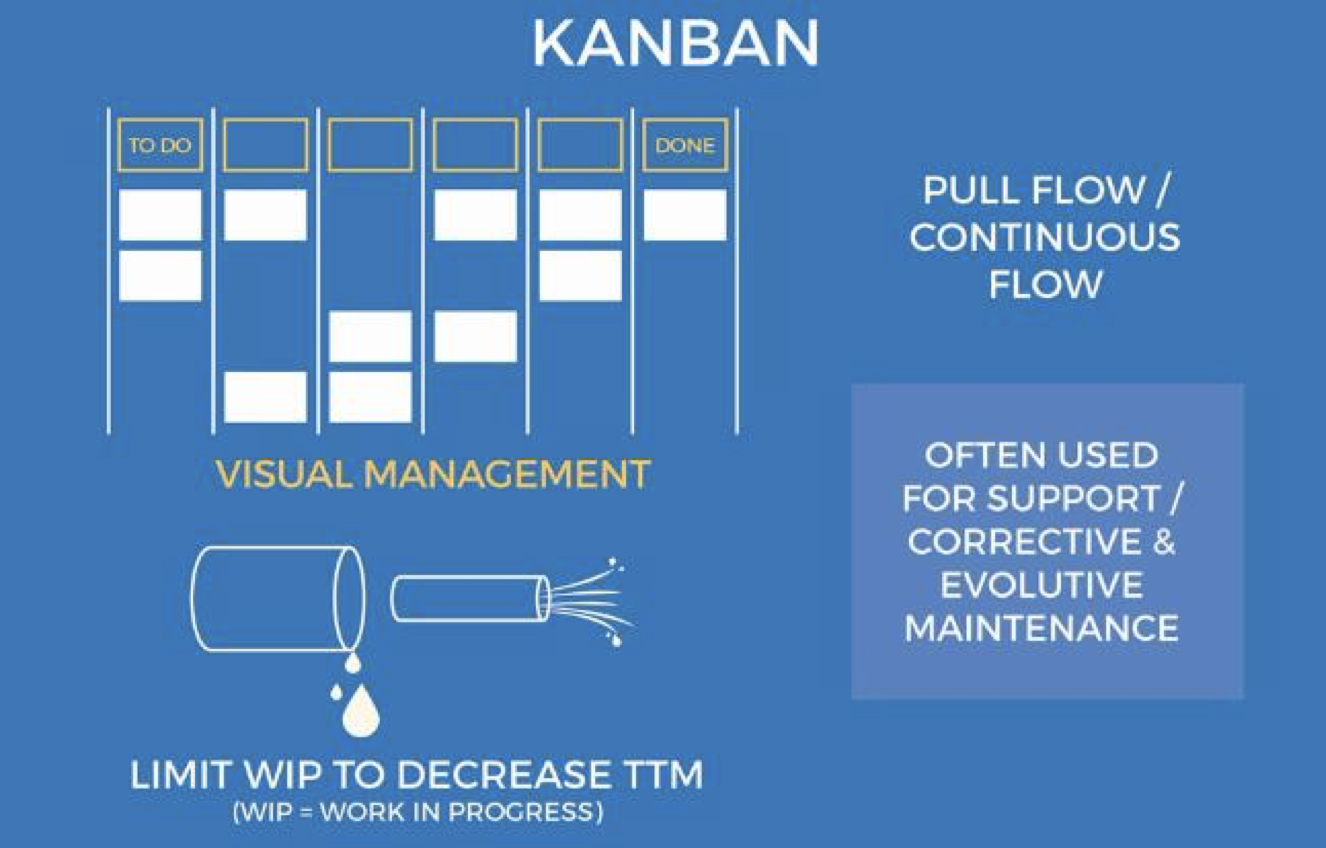


Figure 7:

Scrumban

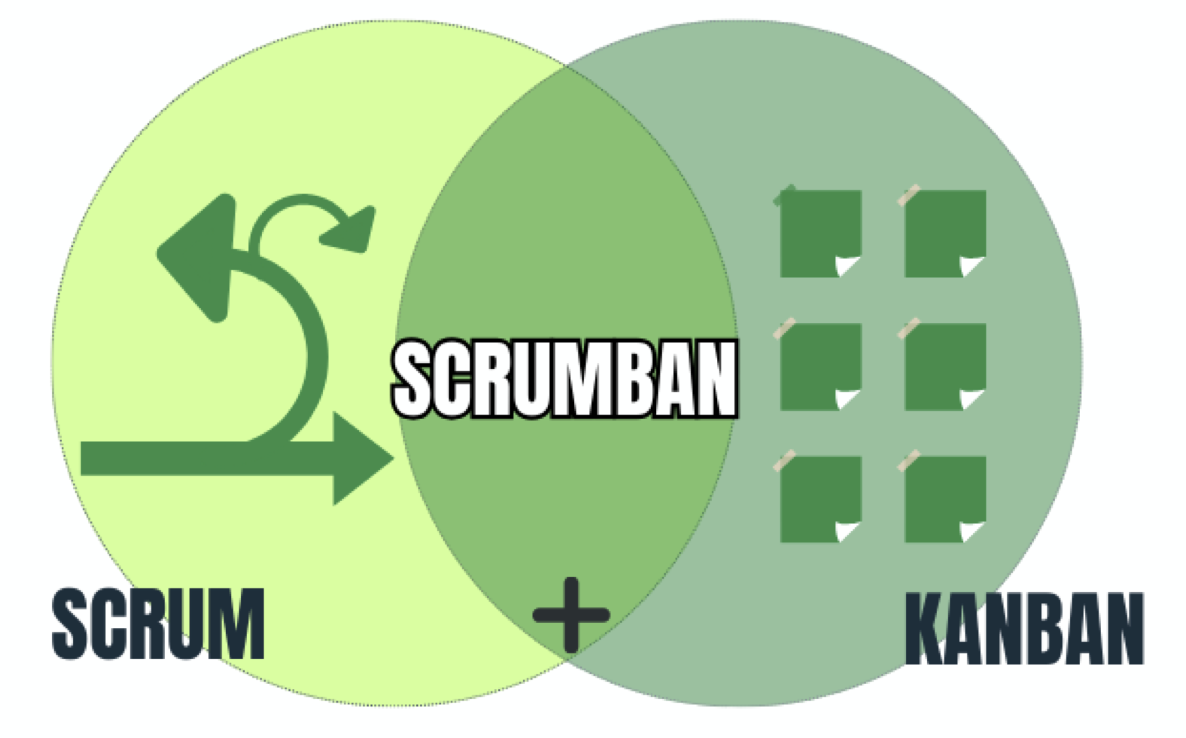
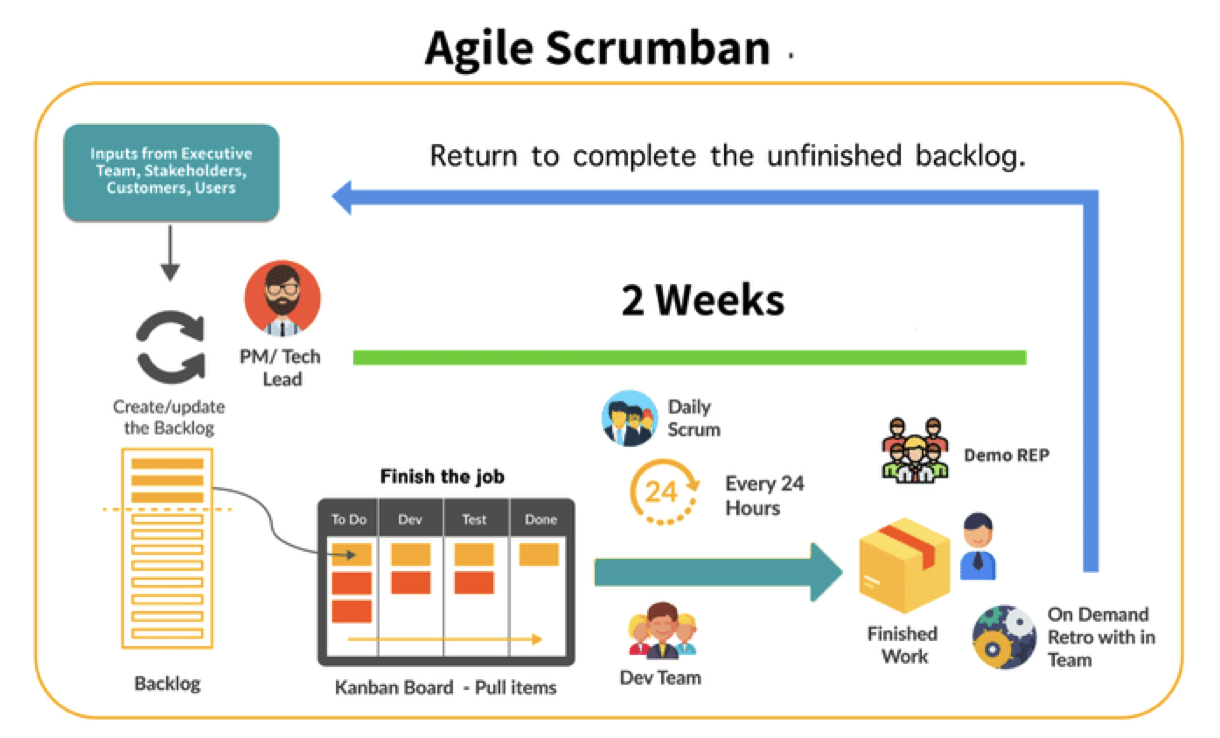


Figure 8:

Agile Scrumban



The project will initially collect requirements to define the product vision, priorities,

and expected outputs. By implementing many sprints, the development lifecycle will be

separated into iterations with the aim of improving the order of priorities for each sprint. We

will create a backlog and establish a sprint target at the beginning of each sprint during the

Sprint Planning Meeting.

Throughout each sprint, user stories will be placed on a Kanban board, with different

columns for tracking an item's progress from backlog to the completed state. We will share

updates, plan the next day, and talk about any obstacles at the daily scrum meetings, which

will last about 15 minutes. At the end of each sprint, we will review a demo of the latest

increment to increase the effectiveness of the next sprint.

By using the Scrumban Agile framework, that combines the iterative development of

Scrum with the continuous flow principles of Kanban, we will assure a collaborative

approach with regular user demos and priority-based progress tracking. The successful and timely completion of the project will be made possible by this process.

# **Tools and Technology**

Here are the different tools and technologies that we will be using during the

development process:

* Windows and MacOS as OS,
* VS Code as IDE,
* MS Teams, WhatsApp and Gmail for discussion,
* Figma, Balsamiq for prototyping/wireframing,
* Java, Java Fx, Java Swing for creating main GUI,
* FXML and CSS for extra ui,
* Gradle build tool for packaging, building and running app,
* JavaWebView for small browser engine for rendreing web pages
* MYSQL Workbench for database storage and management,
* Visual Paradigm Online conceptual diagram,
* MS Word, MS Excel, Google Docs, Google Sheets for keeping sprint records, documentation, Gantt charts and more,
* Canva for creating graphics,
* Git and GitHub for version control,
* Google and YouTube as a primary source of research

Figure 9:

Tools and technologies used

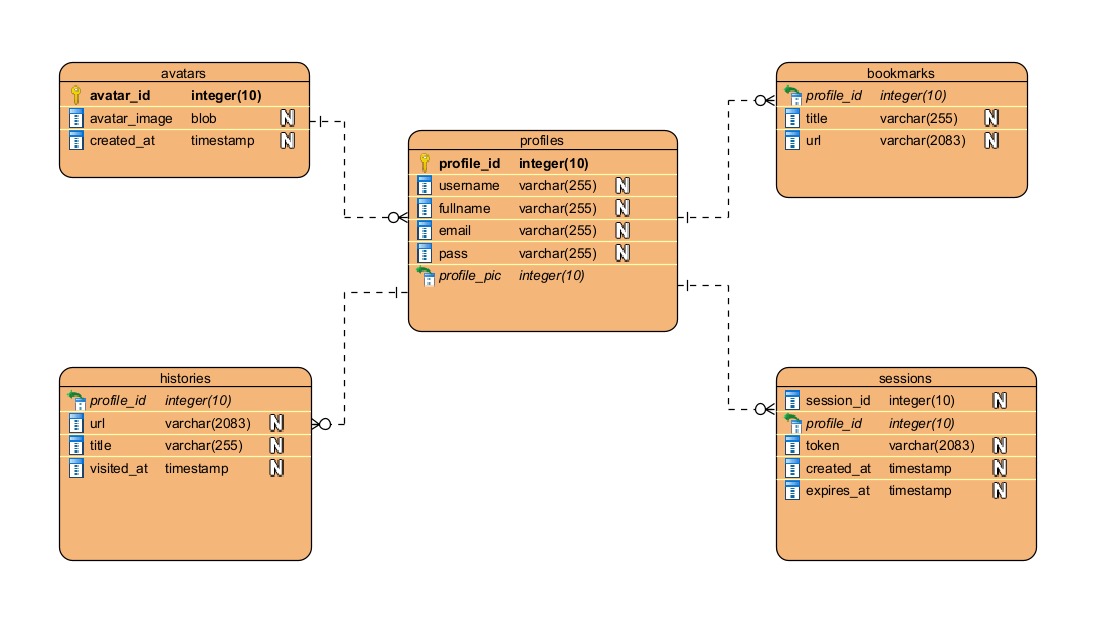


# **Conceptual diagram**

A graphical representation of a database that depicts entities, relationships, and attributes within a system is called an entity-relationship (ER) diagram. The project’s ER diagram is shown in the following figure.

Figure 10:

ER Diagram



# **Prototype**

# **Scope**

Project scope is determined by combining all the previously discussed topics,

including aim, objectives, problems, features, FRs, and NFRs. It contains features included,

excluded within the project, limitations & boundaries of the project, and more.

Included Features/In-Scope

* Unique profiles with proper management of history and bookmarks according to the preference of the user.
* Dark mode theme customization as per user’s preference.
* Custom title bar to minimize, maximize or close
* Bookmark feature for saving the links across profiles.
* History management to facilitate past opened links efficiently.
* Modular design for a sleek browsing experience

Limitations/Out-of-scope

* Voice or gesture navigation cannot work as an input for searching.
* UI is not designed to handle mobile devices.
* No parental control settings.

# **SWOT Analysis**

Figure 11:

SWOT Analysis



# **Conclusion**

In conclusion, Evoza aims to reshape the browsing experience focusing on privacy, customizability, and efficiency. Designed with features such as multi-user profiles, Guest/ Incognito Modes, personalized themes, and modular architecture it suits the needs of varied users today with an additional layer of customization and security. Evoza is a response to the limitations of traditional browsers and the evolution of user-centric innovations, and they are creating a seamless, adaptable, and ever-evolving browsing experience for the personal, professional, and shared context of everyday lives. Its emphasis on performance, usability and security positions makes it a reliable tool for navigating the complexities of the digital world, empowering users to browse with confidence and control.

# **References**

Kumar, A. (2024, September 30). *How technology changed our lives: Has it improved life today?* Simplilearn.com. <https://www.simplilearn.com/how-has-technology-improved-our-lives-article>

Olsen, G. (2022, August 8). *Comparing web browser privacy and security features*. Search Enterprise Desktop. <https://www.techtarget.com/searchenterprisedesktop/tip/Comparing-web-browser-privacy-and-security-features>

GeeksforGeeks. (2024, November 11). *Best Web Browsers: Top picks for PC, laptop, Chromebook, Windows, and Mac devices*. GeeksforGeeks. <https://www.geeksforgeeks.org/browsers-for-pc/>

Sean. (2024, December 3). Vivaldi: The browser that reinvents the web experience | DebugBar. *DebugBar*. <https://www.debugbar.com/vivaldi-the-browser-that-reinvents-the-web-experience/>

Gate Smashers. (2022, July 14). *Functional vs Non-functional Requirements | Requirement Engineering | Software Engineering*[Video]. YouTube. <https://www.youtube.com/watch?v=IBqO6aUkJSE>

Laoyan, S. (2024, February 2). What is Agile Methodology? (A Beginner’s Guide) [2024] • Asana. *Asana*. <https://asana.com/resources/agile-methodology>

Hitesh Choudhary. (2023, July 25).*How does a browser work? | Engineering side*[Video]. YouTube. <https://youtu.be/5rLFYtXHo9s?si=pMcrBL31Gc5f7S9r>